Is sentence compression an NLG task?

Marsi, E.C.; Krahmer, E.J.; Hendrickx, I.; Daelemans, W.

Published in:
Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)

Publication date:
2009

Is sentence compression an NLG task?

Erwin Marsi, Emiel Krahmer
Tilburg University
Tilburg, The Netherlands
e.j.krahmer@uvt.nl
e.c.marsi@uvt.nl

Iris Hendrickx, Walter Daelemans
Antwerp University
Antwerpen, Belgium
iris.hendrickx@ua.ac.be
walter.daelemans@ua.ac.be

Abstract

Data-driven approaches to sentence compression define the task as dropping any subset of words from the input sentence while retaining important information and grammaticality. We show that only 16% of the observed compressed sentences in the domain of subtitling can be accounted for in this way. We argue that part of this is due to evaluation issues and estimate that a deletion model is in fact compatible with approximately 55% of the observed data. We analyse the remaining problems and conclude that in those cases word order changes and paraphrasing are crucial, and argue for more elaborate sentence compression models which build on NLG work.

1 Introduction

The task of *sentence compression* (or *sentence reduction*) can be defined as summarizing a single sentence by removing information from it (Jing and McKeown, 2000). The compressed sentence should retain the most important information and remain grammatical. One of the applications is in automatic summarization in order to compress sentences extracted for the summary (Lin, 2003; Jing and McKeown, 2000). Other applications include automatic subtitling (Vandeghinste and Tsjong Kim Sang, 2004; Vandeghinste and Pan, 2004; Daelemans et al., 2004) and displaying text on devices with very small screens (Corston-Oliver, 2001).

A more restricted version defines sentence compression as dropping any subset of words from the input sentence while retaining important information and grammaticality (Knight and Marcu, 2002). This formulation of the task provided the basis for the noisy-channel en decision-tree based algorithms presented in (Knight and Marcu, 2002), and for virtually all follow-up work on data-driven sentence compression (Le and Horiguchi, 2003; Vandeghinste and Pan, 2004; Turner and Charniak, 2005; Clarke and Lapata, 2006; Zajic et al., 2007; Clarke and Lapata, 2008) It makes two important assumptions: (1) only word deletions are allowed – no substitutions or insertions – and therefore no paraphrases; (2) the word order is fixed. In other words, the compressed sentence must be a *subsequence* of the source sentence. We will call this *the subsequence constraint*, and refer to the corresponding compression models as *word deletion models*. Another implicit assumption in most work is that the scope of sentence compression is limited to isolated sentences and that the textual context is irrelevant.

Under this definition, sentence compression is reduced to a word deletion task. Although one may argue that even this counts as a form of text-to-text generation, and consequently an NLG task, the generation component is virtually non-existent. One can thus seriously doubt whether it really is an NLG task.

Things would become more interesting from an NLG perspective if we could show that sentence compression necessarily involves transformations beyond mere deletion of words, and that this requires linguistic knowledge and resources typical to NLG. The aim of this paper is therefore to challenge the deletion model and the underlying subsequence constraint. To use an analogy, our aim is to show that sentence compression is less like carving something out of wood - where material can only be removed - and more like molding something out of clay - where the material can be thor-

Proceedings of the 12th European Workshop on Natural Language Generation, pages 25–32,
Athens, Greece, 30 – 31 March 2009.

oughly reshaped. In support of this claim we provide evidence that the coverage of deletion models is in fact rather limited and that word reordering and paraphrasing play an important role.

The remainder of this paper is structured as follows. In Section 2, we introduce our text material which comes from the domain of subtitling. We explain why not all material is equally well suited for studying sentence compression and motivate why we disregard certain parts of the data. We also describe the manual alignment procedure and the derivation of edit operations from it. In Section 3, an analysis of the number of deletions, insertions, substitutions, and reorderings in our data is presented. We determine how many of the compressed sentences actually satisfy the subsequence constraint, and how many of them could in principle be accounted for. That is, we consider alternatives with the same compression ratio which do not violate the subsequence constraint. Next is an analysis of the remaining problematic cases in which violation of the subsequence constraint is crucial to accomplish the observed compression ratio. We single out (1) reordering after deletion and (2) paraphrasing as important factors. Given the importance of paraphrases, Section 3.4 discusses the perspectives for automatic extraction of paraphrase pairs from large text corpora, and tries to estimate how much text is required to obtain a reasonable coverage. We finish with a summary and discussion in Section 4.

2 Material

We study sentence compression in the context of subtitling. The basic problem of subtitling is that on average reading takes more time than listening, so subtitles can not be a verbatim transcription of the speech without increasingly lagging behind. Subtitles can be presented at a rate of 690 to 780 characters per minute, while the average speech rate is considerably higher (Vandeghinste and Tsjong Kim Sang, 2004). Subtitles are therefore often a compressed representation of the original spoken text.

Our text material stems from the *NOS Journaal*, the daily news broadcast of the Dutch public television. It is parallel text with on one side the *autocue* sentences (aut), i.e. the text the news reader is reading, and on the other side the corresponding *subtitle* sentences (sub). It was originally collected and processed in two earlier research projects – Atranos and Musa – on automatic subtitling (Vandeghinste and Tsjong Kim Sang, 2004; Vandeghinste and Pan, 2004; Daelemans et al., 2004). All text was automatically tokenized and aligned at the sentence level, after which alignments were manually checked.

The same material was further annotated in an ongoing project called DAESO[1], in which the general goal is automatic detection of semantic overlap. All aligned sentences were first syntactically parsed after which their parse trees were manually aligned in more detail. Pairs of similar syntactic nodes – either words or phrases – were aligned and labeled according to a set of five semantic similarity relations (Marsi and Krahmer, 2007). For current purposes, only the alignment at the word level is used, ignoring phrasal alignments and relation labels.

Not all material in this corpus is equally well suited for studying sentence compression as defined in the introduction. As we will discuss in more detail below, this prompted us to disregard certain parts of the data.

Sentence deletion, splitting and merging For a start, autocue and subtitle sentences are often not in a one-to-one alignment relation. Table 1 specifies the alignment degree (i.e. the number of other sentences that a sentence is aligned to) for autocue and subtitle sentences. The first thing to notice is that there is a large number of unaligned subtitles. These correspond to non-anchor text from, e.g., interviews or reporters abroad. More interesting is that about one in five autocue sentences is completely dropped. A small number of about 4 to 8 percent of the sentence pairs are not one-to-one aligned. A long autocue sentence may be split into several simpler subtitle sentences, each containing only a part of the semantic content of the autocue sentence. Conversely, one or more - usually short - autocue sentences may be merged into a single subtitle sentence.

These decisions of sentence deletion, splitting and merging are worthy research topics in the context of automatic subtitling, but they should not be confused with sentence compression, the scope of which is by definition limited to single sentence. Accordingly we disregarded all sentence pairs where autocue and subtitle are not in a one-to-one relation with each other. This reduced the data set from 15289 to 11034 sentence pairs.

[1] `http://daeso.uvt.nl`

Degree:	Autocue:	(%)	Subtitle:	(%)
0	3607	(20.74)	12542	(46.75)
1	12382	(71.19)	13340	(49.72)
2	1313	(7.55)	901	(3.36)
3	83	(0.48)	41	(0.15)
4	8	(0.05)	6	(0.02)

Table 1: Degree of sentence alignment

Word compression A significant part of the reduction in subtitle characters is actually not obtained by deleting words but by lexical substitution of a shorter token. Examples of this include substitution by digits ("7" for "seven"), abbreviations or acronyms ("US" for "United States"), symbols (euro symbol for "Euro"), or reductions of compound words ("elections" for "state-elections"). We will call this *word compression*. Although an important part of subtitling, we prefer to abstract from word compression and focus here on sentence compression proper. Removing all sentence pairs containing a word compression has the disadvantage of further reducing the data set. Instead we choose to measure *compression ratio* (CR) in terms of tokens[2] rather than characters.

$$CR = \frac{\#tok_{sub}}{\#tok_{aut}} \quad (1)$$

This means that the majority of the word compressions do not affect the sentence CR.

Variability in compression ratio The CR of subtitles is not constant, but varies depending (mainly) on the amount of provided autocue material in a given time frame. The histogram in Figure 1 shows the distribution of the CR (measured in words) for one-to-one aligned sentences. In fact, autocue sentences are most likely not to be compressed at all (thus belonging to the largest bin, from 1.00 to 1.09 in the histogram).[3] In order to obtain a proper set of compression examples, we retained only those sentence pairs where the compression ratio is less than one.

Parsing failures As mentioned earlier detailed alignment of autocue and subtitle sentences was carried out on their syntactic trees. However, for various reasons a small number of sentences (0.2%) failed to pass the parser and received no parse tree. As a consequence, their trees could not be aligned and there is no alignment at the word level available either. Variability in CR and parsing failures are together responsible for a further reduction down to 5233 sentence pairs, the final size of our data set, with an overall CR of 0.69. Other properties of this data set are summarized in Table 2.[4]

[2]Throughout this study we ignore punctuation and letter case.

[3]Some instances even show a CR larger than one, because occasionally there is sufficient time/space to provide a clarification, disambiguation, update, or stylistic enhancement.

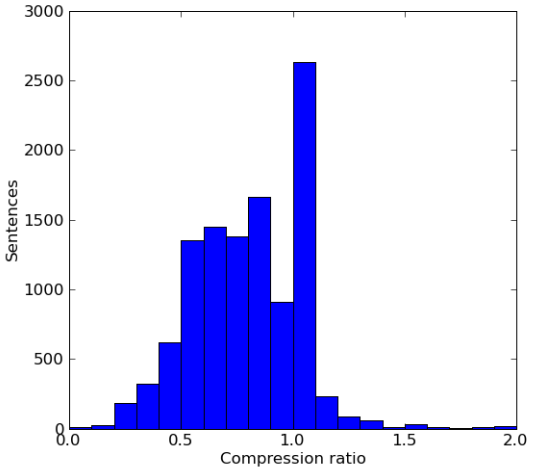

Figure 1: Histogram of compression ratio

	Min:	Max:	Sum:	Mean:	SD:
aut-tokens	2	43	80651	15.41	5.48
sub-tokens	1	29	53691	10.26	3.72
CR	0.07	0.96	nan	0.69	0.17

Table 2: Properties of the final data set of 5233 pairs of autocue-subtitle sentences: minimum value, maximal value, total sum, mean and standard deviation for number of tokens per autocue/subtitle sentence and Compression Ratio

Word deletions, insertions and substitutions Having a manual alignment of similar words in both sentences allows us to simply deduce word deletions, substitutions and insertions, as well as word order changes, in the following way:

- if an autocue word is not aligned to a subtitle word, then it is was deleted
- if a subtitle word is not aligned to an autocue word, then it was inserted
- if different autocue and subtitle words are aligned, then the former was substituted by the latter
- if alignments cross each other, then the word order was changed

The remaining option is where the aligned words are identical (ignoring differences in case).

[4]We use the acronym *nan* ("not a number") for undefined/meaningless values.

Without the word alignment, we would have to resort to automatically calculating the edit distance, i.e. the sum of the minimal number of insertions, deletions and substitutions required to transform one sentence in to the other. However, this would result in different and often counter-intuitive sequences of edit operations. Our approach clearly distinguishes word order changes from the edit operations; the conventional edit distance, by contrast, can only account for changes in word order by sequences of the edit operations. Another difference is that substitution can also be accomplished as deletion followed by insertion, which means edit operations need to have an associated weight. Global tuning of these weights turns out to be hard.

3 Analysis

3.1 Edit operations

The observed deletions, insertions, substitutions, edit distances, and word order changes are shown in Table 3. As expected, deletion is the most frequent operation, with on average seven deletions per sentence. Insertion and substitutions are far less frequent. Note also that – even though the task is compression – insertions are somewhat more frequent than substitutions. Word order changes occur in 1688 cases (32.26%). Here, reordering is a binary variable – i.e. the word order is changed or not – hence Min, Max and SD are undefined.

Another point of view is to look at the number of sentence pairs containing a certain edit operation. Here we find 5233 pairs (100.00%) with deletion, 2738 (52.32%) with substitution, 3263 (62.35%) with insertion, and 1688 (32.26%) with reordering.

The average CR for subsequences is 0.68 ($SD = 0.20$) versus 0.69 ($SD = 0.17$) for non-subsequences. A detailed inspection of the relation between the $subsequence/non-subsequence$ ratio and CR revealed no clear correlation, so we did not find indications that non-subsequences occur more frequently at higher compression ratios.

3.2 Percentage of subsequences

The subtitle is a subsequence of the autocue if there are no insertions, no substitutions, and no word order changes. In contrast, if any of these do occur, the subtitle is not a subsequence. It turns out that only 843 (16.11%) subtitles are a subsequence, which is rather low.

	Min:	Max:	Sum:	Mean:	SD:
del	1	34	34728	6.64	4.57
sub	0	6	4116	0.79	0.94
ins	0	17	7768	1.48	1.78
dist	1	46	46612	8.91	5.78
reorder	nan	nan	1688	0.32	nan

Table 3: Observed word deletions, insertions, substitutions, and edit distances

At first sight, this appears to be bad news for any deletion model, as it seems to imply that the model cannot account for close to 84% the observed data. However, the important thing to keep in mind is that compression of a given sentence is a problem for which there are usually multiple solutions (Belz and Reiter, 2006). This is exactly what makes it so hard to perform automatic evaluation of NLG systems. There may very well exist semantically equivalent alternatives with the same CR which do satisfy the subsequence constraint. For this reason, a substantial part of the observed non-subsequences may have subsequence counterparts which can be accounted for by a deletion model. The question is: how many?

In order to address this question, we took a random sample of 200 non-subsequence sentence pairs. In each case we tried to come up with an alternative subsequence subtitle with the same meaning and the same CR (or when opportune, even a lower CR). Table 4 shows the distribution of the difference in tokens between the original non-subsequence subtitle and the manually-constructed equivalent subsequence subtitle. Apparently 95 out of 200 (47%) subsequence subtitles have the same (or even fewer) tokens, and thus the same (or an even lower) compression ratio. This suggests that the subsequence constraint is not as problematic as it seemed and that the coverage of a deletion model is in fact far better than it appeared to be. Recall that 16% of the original subtitles were already subsequences, so our analysis suggests that a deletion model is compatible with 55% (16% plus 47% of 84%).

3.3 Problematic non-subsequences

Another result of this exercise in rewriting subtitles is that it allows us to identify those cases where the attempt to create a proper subsequence fails. In (1), we show one representative example of a problematic subtitle, for which

(1) **Aut** de bron was een geriatrische patient die zonder het zelf te merken uitzonderlijk veel larven bij zich
the source was a geriatric patient who without it self to notice exceptionally many larvae with him
bleek te dragen en een grote verspreiding veroorzaakte
appeared to carry and a large spreading caused
"the source was a geriatric patient who unknowingly carried exceptionally many larvae and caused a wide spreading"

Sub een geriatrische patient met larven heeft de verspreiding veroorzaakt
a geriatric patient with larvae has the spreading caused
Seq de bron was een geriatrische patient die veel larven bij zich bleek te dragen en een verspreiding veroorzaakte

(2) **Aut** in verband met de lawineramp in galür hebben de politieke partijen in tirol gezamenlijk besloten de
in relation to the avalanche-disaster in Galtür have the political parties in Tirol together decided the
verkiezingscampagne voor het regionale parlement op te schorten
election-campaign for the regional parliament up to postpone
Sub de politieke partijen in tirol hebben besloten de verkiezingen op te schorten
the political parties in Tirol have decided the elections up to postpone
"Political parties in Tirol have decided to postpone the elections"

(3) **Aut** velen van hen worden door de serviërs in volgeladen treinen gedeporteerd
many of them are by the Serbs in crammed trains deported
Sub vluchtelingen worden per trein gedeporteerd
refugees are by train deported

token-diff:	count:	(%:)
-2	4	2.00
-1	18	9.00
0	73	36.50
1	42	21.00
2	32	16.00
3	11	5.50
4	9	4.50
5	5	2.50
7	2	1.00
8	2	1.00
9	1	0.50
11	1	0.50

Table 4: Distribution of difference in tokens between original non-subsequence subtitle and equivalent subsequence subtitle

the best equivalent subsequence we could obtain still has nine more tokens than the original non-subsequence. These problematic non-subsequences reveal where insertion, substitution and/or word reordering are essential to obtain a subtitle with a sufficient CR (i.e. the CR observed in the real subtitles). At least three different types of phenomena were observed.

Word order In some cases deletion of a constituent necessitates a change in word order to obtain a grammatical sentence. In example (2), the autocue sentence has the PP modifier *in verband met de lawineramp in galür* in its topic position (first sentence position). Deleting this modifier, as is done in the subtitle, results in a sentence that starts with the verb *hebben*, which is interpreted as a yes-no question. For a declarative interpretation, we have to move the subject *de politieke partijen* to the first position, as in the subtitle. Incidentally, this indicates that it is instructive to apply sentence compression models to multiple languages, as a word order problem like this never arises in English.

Similar problems arise whenever an embedded clause is promoted to a main clause, which requires a change in the position of the finite verb in Dutch. In total, a word order problem occurred in 24 out 200 sentences.

Referring expressions Referring expressions are on many occasions replaced by a shorter one – usually a little less precise. For example, *de belgische overheid* 'the Belgian authorities' is replaced by *belgie* 'Belgium'. Extreme cases of this occur where a long NP like *deze tweede impeachment-procedure in de amerikaanse geschiedenis* 'this second impeachment-procedure in the American history' is replaced by an anaphor like *het* 'it'.

Since a referring expression or anaphor must be appropriate in the given context, substitutions like these transcend the domain of a single sentence and require taking the preceding textual context into account. This is especially clear in examples like (3) in which 'many of them' is replaced the 'refugees'. It is questionable whether these types of substitutions belong to the task of sentence compression. We prefer to regard it as one of the additional tasks in automatic subtitling, apart from compression. Incidentally, it is interesting that the challenge of generating referring expressions is also relevant for automatic subtitling.

Paraphrasing Apart from the reduced referring expressions, there are nominal paraphrases reducing a noun phrases like *medewerkers van banken* 'employees of banks' to a compound word like *bankmedewerkers* 'bank-employees'. Likewise, there are adverbial paraphrases such as *sinds een paar jaar* 'since a few years' to *tegenwoordig* 'nowadays', and *van de afgelopen tijd* 'of the past time' to *recent* 'recent'. However, the majority of the paraphrasing concerns verbs as in the two examples below.

(4) **Aut** X neemt het initiatief tot oprichting van Y
X takes the initiative to raising of Y
Sub X zet Y op
X sets Y up

(5) **Aut** X om zijn uitlevering vroeg maar Y die weigerde
X for his extradition asked but Y that refused
Sub Y hem niet wilde uitleveren aan X
Y him not wanted extradite to Y
"Y refused to extradite him to Y"

Even though not all paraphrases are actually shorter, it seems that at least some of them boost compression beyond what can be accomplished with only word deletion. In the next Section, we look at the possibilities of automatic extraction of such paraphrases.

3.4 Perspectives for automatic paraphrase extraction

There is a growing amount of work on automatic extraction of paraphrases from text corpora (Lin and Pantel, 2001; Barzilay and Lee, 2003; Ibrahim et al., 2003; Dolan et al., 2004). One general prerequisite for learning a particular paraphrase pattern is that it must occur in the text corpus with a sufficiently high frequency, otherwise the chances of learning the pattern are proportionally small. In this section, we investigate to what extent the paraphrases encountered in our random sample of 200 pairs can be retrieved from a reasonably large text corpus.

In a first step, we manually extracted 106 paraphrase patterns. We filtered these patterns and excluded anaphoric expressions, general verb alternation patterns like active/passive and continuous/non-continuous, as well as verbal patterns involving more than two slots. After this filtering step, 59 pairs of paraphrases remained, including the examples shown in the preceding Section.

The aim was to estimate how big our corpus has to be to cover the majority of these paraphrase pairs. We started with counting for each of the paraphrase pairs in our sample how often they occur in a corpus of Dutch news texts, the Twente News Corpus[5], which contains approximately 325M tokens and 20M sentences. We employed regular expressions to count the number of paraphrase pattern matches. The corpus turned out to contain 70% percent of all paraphrase pairs (i.e. both patterns in the pair occur at least once). We also counted how many pairs have a frequencies of at least 10 and 100. To study the effect of corpus size on the percentage of covered paraphrases, we performed these counts on 1, 2, 5, 10, 25, 50 and 100% of the corpus. Figure 2 shows the percentage of covered paraphrases dependent on the corpus size. The most strict threshold that only counts pairs that occur at least 100 times in our corpus, does not retrieve any counts on 1% of the corpus (3M words). At 10% of the corpus size only 4% of the paraphases is found, and on the full data set 25% of the pairs is found.

For 51% percent of the patterns (with a frequency of at least 10) we find substantial evidence in our corpus of 325M tokens. We fitted a curve through our data points, and found a logarithmic line fit with adjusted R^2 value of .943. This suggests that in order to get 75% of the patterns, we would need a corpus that is 18 times bigger than our current one, which amounts to roughly 6 billion words. Although this seems like a lot of text, using the WWW as our corpus would easily give us these numbers. Today's estimate of the Index Dutch World Wide Web is 688 million pages[6]. If we assume that each page contains at least 100 tokens on average, this implies a corpus size of 68 billion tokens.

The patterns used here are word-based and in many cases they express a particular verb tense or verb form (e.g. 3rd person singular), and word order. This implies that our estimations are the minimum number of matches one can find. For more abstract matching, we would need syntactically parsed data (Lin and Pantel, 2001). We expect that this would also positively affect the coverage.

[5] http://www.vf.utwente.nl/~druid/TwNC/TwNC-main.html

[6] http://www.worldwidewebsize.com/index.php?lang=NL, as measured in December 2008

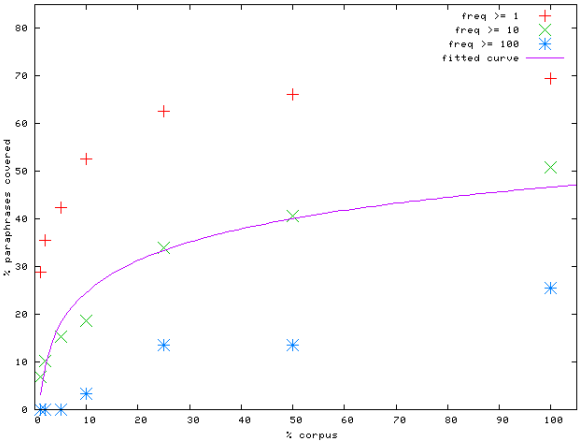

Figure 2: Percentage of covered paraphrases as a function of the corpus size

4 Discussion

We found that only 16.11% of 5233 subtitle sentences were proper subsequences of the corresponding autocue sentence, and therefore 84% can not be accounted for by a deletion model. One consequence appears to be that the subsequence constraint greatly reduces the amount of available training material for any word deletion model. However, an attempt to rewrite non-subsequences to semantically equivalent sequences with the same CR suggests that a deletion model could in principle be adequate for 55% of the data. Moreover, in those cases where an application can tolerate a little slack in the CR, a deletion model might be sufficient. For instance, if we are willing to tolerate up to two more tokens, we can account for as much as 169 (84%) of the 200 non-subsequences in our sample, which amounts to 87% (16% plus 84% of 84%) of the total data.

It should be noted that we have been very strict regarding what counts as a semantically equivalent subtitle: every piece of information occurring in the non-subsequence subtitle must reoccur in the sequence subtitle. However, looking at our original data, it is clear that considerable liberty is taken as far as conserving semantic content is concerned: subtitles often drop substantial pieces of information. If we relax the notion of semantic equivalence a little, an even larger part of the non-subsequences can be rewritten as proper sequences.

The remaining problematic non-subsequences are those where insertion, substitution and/or word reordering are essential to obtain a sufficient CR. One of the issues we identified is that deletion of certain constituents must be accompanied by a change in word order to prevent an ungrammatical sentence. Since changes in word order appear to require grammatical modeling or knowledge, this brings sentence compression closer to being an NLG task.

Nguyen and Horiguchi (2003) describe an extension of the decision tree-based compression model (Knight and Marcu, 2002) which allows for word order changes. The key to their approach is that dropped constituents are temporarily stored on a *deletion stack*, from which they can later be re-inserted in the tree where required. Although this provides an unlimited freedom for rearranging constituents, it also complicates the task of learning the parsing steps, which might explain why their evaluation results show marginal improvements at best.

In our data, most of the word order changes appear to be minor though, often only moving the verb to second position after deleting a constituent in the topic position. We believe that unrestricted word order changes are perhaps not necessary and that the vast majority of the word order problems can be solved by a fairly restricted way of reordering. In particular, we plan to implement a parser-based model with an additional swap operation that swaps the two topmost items on the stack. We expect that this is more feasible as a learning task than a model with a deletion stack.

Apart from reordering, other problems for word deletion models are the insertions and substitutions as a result of paraphrasing. Within a decision tree-based model, paraphrasing of words or continuous phrases may be modeled by a combination of a paraphrase lexicon and an extra operation which replaces the n topmost elements on the stack by the corresponding paraphrase. However, paraphrases involving variable arguments, as typical for verbal paraphrases, cannot be accounted for in this way. More powerful compression models may draw on existing NLG methods for text revision (Inui et al., 1992) to accommodate full paraphrasing.

We also looked at the perspectives for automatic paraphrase extraction from large text corpora. About a quarter of the required paraphrase patterns was found at least a hundred times in our corpus of 325M tokens. Extrapolation suggests that using the web at its current size would give us a coverage of approximately ten counts for three

quarters of the paraphrases.

Incidentally, we identified two other tasks in automatic subtitling which are closely related to NLG. First, splitting and merging of sentences (Jing and McKeown, 2000), which seems related to content planning and aggregation. Second, generation of a shorter referring expression or an anaphoric expression, which is currently one of the main themes in data-driven NLG.

In conclusion, we have presented evidence that deletion models for sentence compression are not sufficient, and that more elaborate models involving reordering and paraphrasing are required, which puts sentence compression in the field of NLG.

Acknowledgments

We would like to thank Nienke Eckhardt, Paul van Pelt, Hanneke Schoormans and Jurry de Vos for the corpus annotation work, and Erik Tsjong Kim Sang and colleagues for the autocue-subtitle material from the ATRANOS project, and Martijn Goudbeek for help with curve fitting. This work was conducted within the DAESO project funded by the Stevin program (De Nederlandse Taalunie).

References

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 16–23, Morristown, NJ, USA.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 313–320.

James Clarke and Mirella Lapata. 2006. Models for sentence compression: a comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 377–384, Morristown, NJ, USA.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression an integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Simon Corston-Oliver. 2001. Text compaction for display on very small screens. In *Proceedings of the Workshop on Automatic Summarization (WAS 2001)*, pages 89–98, Pittsburgh, PA, USA.

Walter Daelemans, Anita Höthker, and Erik Tjong Kim Sang. 2004. Automatic sentence simplification for subtitling in Dutch and English. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1045–1048.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Morristown, NJ, USA.

Ali Ibrahim, Boris Katz, and Jimmy Lin. 2003. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the 2nd International Workshop on Paraphrasing*, volume 16, pages 57–64, Sapporo, Japan.

Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. 1992. Text Revision: A Model and Its Implementation. In *Proceedings of the 6th International Workshop on Natural Language Generation: Aspects of Automated Natural Language Generation*, pages 215–230. Springer-Verlag London, UK.

Hongyan Jing and Kathleen McKeown. 2000. Cut and paste based text summarization. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, pages 178–185, San Francisco, CA, USA.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

Nguyen Minh Le and Susumu Horiguchi. 2003. A New Sentence Reduction based on Decision Tree Model. In *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation*, pages 290–297.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Chin-Yew Lin. 2003. Improving summarization performance by sentence compression - A pilot study. In *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*, volume 2003, pages 1–9.

Erwin Marsi and Emiel Krahmer. 2007. Annotating a parallel monolingual treebank with semantic similarity relations. In *Proceedings of the 6th International Workshop on Treebanks and Linguistic Theories*, pages 85–96, Bergen, Norway.

Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 290–297, Ann Arbor, Michigan, June.

Vincent Vandeghinste and Yi Pan. 2004. Sentence compression for automated subtitling: A hybrid approach. In *Proceedings of the ACL Workshop on Text Summarization*, pages 89–95.

Vincent Vandeghinste and Erik Tsjong Kim Sang. 2004. Using a Parallel Transcript/Subtitle Corpus for Sentence Compression. In *Proceedings of LREC 2004*.

David Zajic, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing Management*, 43(6):1549–1570.